BAB III

ANALISA DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan tentang perancangan sistem dari aplikasi agen cerdas dalam game labirin. Analisa dan perancangan sistem ini bertujuan untuk mengimplementasikan algoritma *A** dan Dijkstra dalam mencari rute optimal dalam game labirin.

3.1 Analisa Sistem

3.1.1 Analisa dan Gambaran Umum

Pencarian jalur terpendek merupakan sebuah cara yang bertujuan agar kita dapat menuju lokasi tertentu secara efisien. Proses pencarian jalur terpendek dilakukan dengan membandingkan setiap kemungkinan jalur yang tersedia sehingga ditemukan sebuah solusi. Dengan kondisi ini akan diterapkan algoritma *A** dan Dijkstra dalam menentukan jalur terpendek.

Algoritma *A** dan Dijkstra merupakan algoritma pencarian jalur terpendek yang bertujuan mengantar kita menuju titik akhir dengan waktu singkat. Algoritma ini mengkalkulasi jarak dan membandingkan setiap kemungkinan hingga didapat solusi dari metode pencarian jalur optimum. Metode ini banyak digunakan pada pencarian rute terpendek pada dunia nyata, namun dalam hal ini algoritma tersebut diimplementasikan dalam game.

Pada *game* ini terdapat petak labirin akan memiliki dinding berwarna merah sebagai penghalang. Dinding-dinding tersebut ditempatkan secara acak, dalam hal ini metode pencarian jalur terpendek akan di terapkan untuk mencari tahu apakah ada jalur yang tersedia antara posisi *Player* dengan posisi target. Jika jalur antara *Player* dan target tersedia maka game siap dimainkan, jika jalur antara *Player* dan target tidak tersedia maka sistem akan mengacak ulang dinding-dinding pada petak labirin hingga ditemukan jalur antara *Player* dan target oleh agen cerdas.

3.1.2 Perbedaan dan Persamaan dengan Penelitian Sebelumnya

Penelitian sebelumnya mengenai labirin adalah petak permainannya yang bersifat statis sehingga ada kemungkinan untuk bisa di hafal, sehingga dalam penelitian ini akan diimplementasikan sebuah petak labirin bersifat dinamis yang

akan berubah secara acak pada setiap permainannya. Penelitian yang akan dilakukan yakni membahas tentang algoritma *A** dan *Dijkstra* sebagai metode pencarian jalur pada petak labirin yang bersifat dinamis.

3.1.3 Analisa Data

Data yang terdapat dalam aplikasi ini terdiri dari 3 macam jenis data yaitu simpul(*node*), A(simpul yang dijalankan), dan nilai dalam jalur. Data simpul yaitu petak-petak kecil sebagai representasi dari area *pathfinding*. Data A merupakan simpul yang sedang dijalankan dalam algortima pencarian jalan terpendek. Data Nilai yaitu nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke A dan H, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan.

3.1.4 Analisa Kebutuhan Sistem

a. Kebutuhan Fungsional

Merupakan kebutuhan yang harus ada pada sistem. Bagaimana sistem bereaksi pada input tertentu dan bagaimana perilaku system pada situasi tertentu.

1) *Hardware* dan *Software*
2) *User interface system*

b. Kebutuhan Non Fungsional

Merupakan kebutuhan yang tidak harus ada pada sistem. Kebutuhan ini menawarkan sistem seperti batasan waktu, batasan pengembangan, dll.

1) Bahasa Pemrograman yang dipakai dalam pembuatan aplikasi.
2) Kebutuhan eksternal.

3.2 Perancangan Sistem

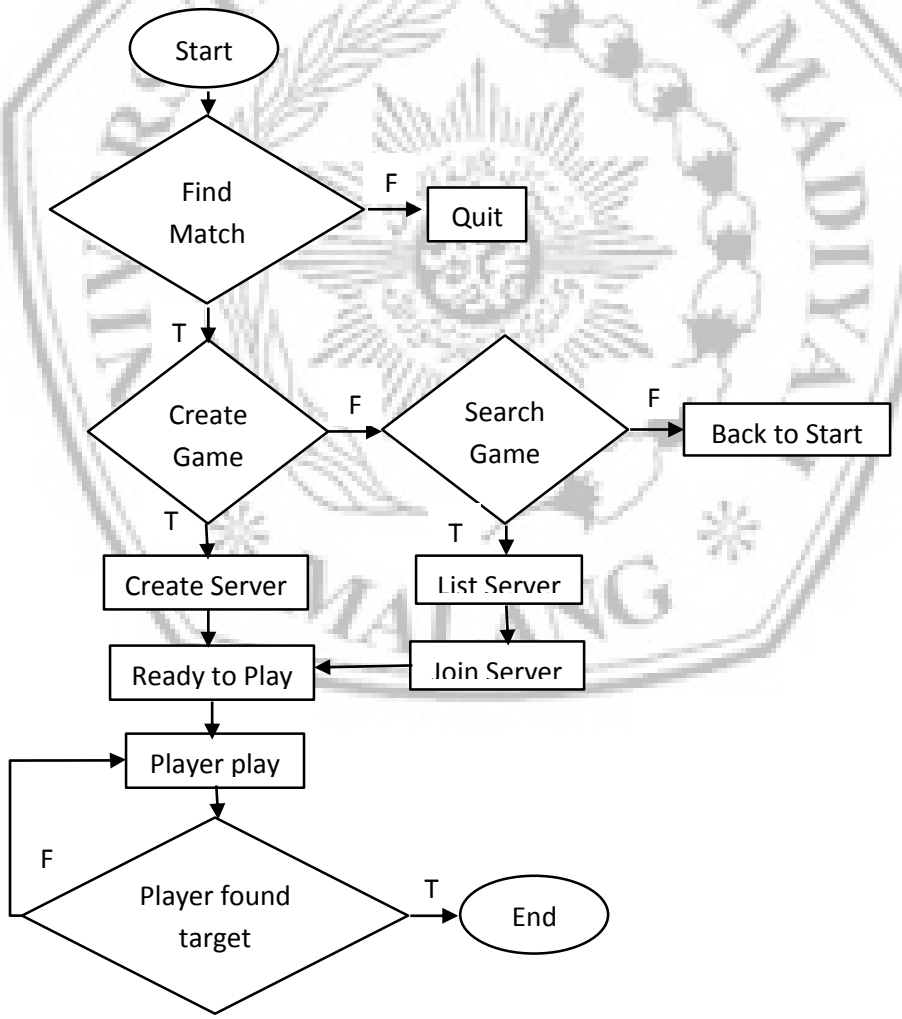
Program ini akan memberikan informasi mengenai algoritma pencarian jalur terpendek. Rancangan kebutuhan-kebutuhan sistem yaitu :

3.2.1 *Flowchart* Sistem

a. Flowchart Game Play

Pembuatan aplikasi agen cerdas ini memiliki menu utama yaitu *Find Match*. Dengan memilih *Find Match* pemain akan diberi pilihan untuk membuat sebuah

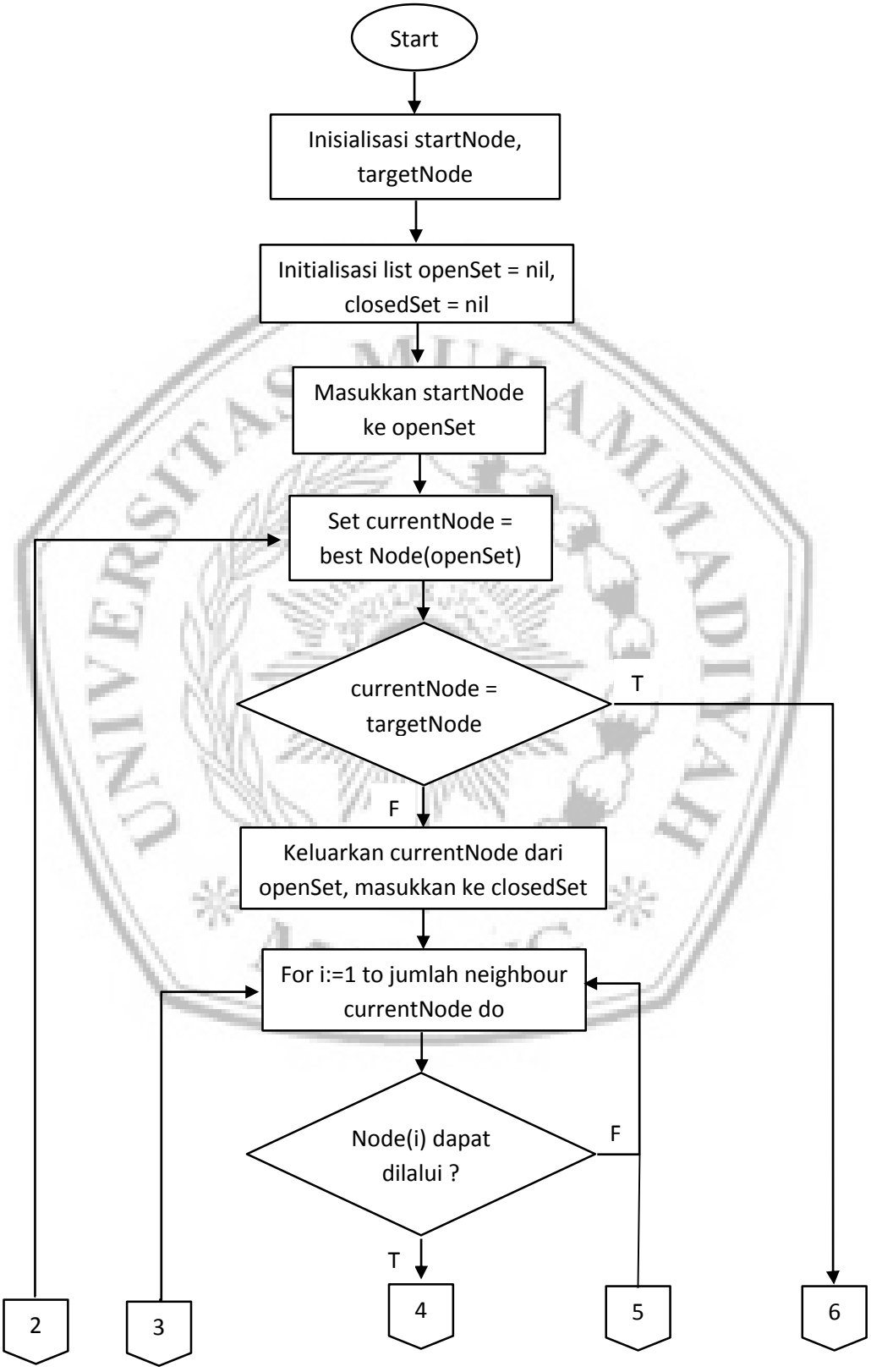
ruang sendiri dan menjadi *server host*, atau memilih *Search Game* untuk mencari *server host* yang sudah tersedia dan bargabung sebagai *client*. Ketika dalam satu *server host* terdapat *server* dan *client* maka *game* secara langsung akan menjalankan algoritma A* untuk menyediakan petak labirin yang akan dimainkan. Petak labirin memiliki dinding penghalang yang disusun secara acak, algoritma A* akan mengecek petak labirin yang telah tersusun dan memberi kesimpulan apakah ada jalur yang tersedia atara karakter pemain dan target. Jika terdapat jalur antara karakter pemain dan target maka game siap dimainkan, jika tidak terdapat jalur antara karakter pemain dan target maka petak labirin akan diacak ulang dan algoritma A* mengecek kembali petak labirin yang baru. Pada level berikutnya berlaku ketentuan yang sama namun menggunakan metode yang berbeda yaitu menggunakan algoritma Dijkstra.

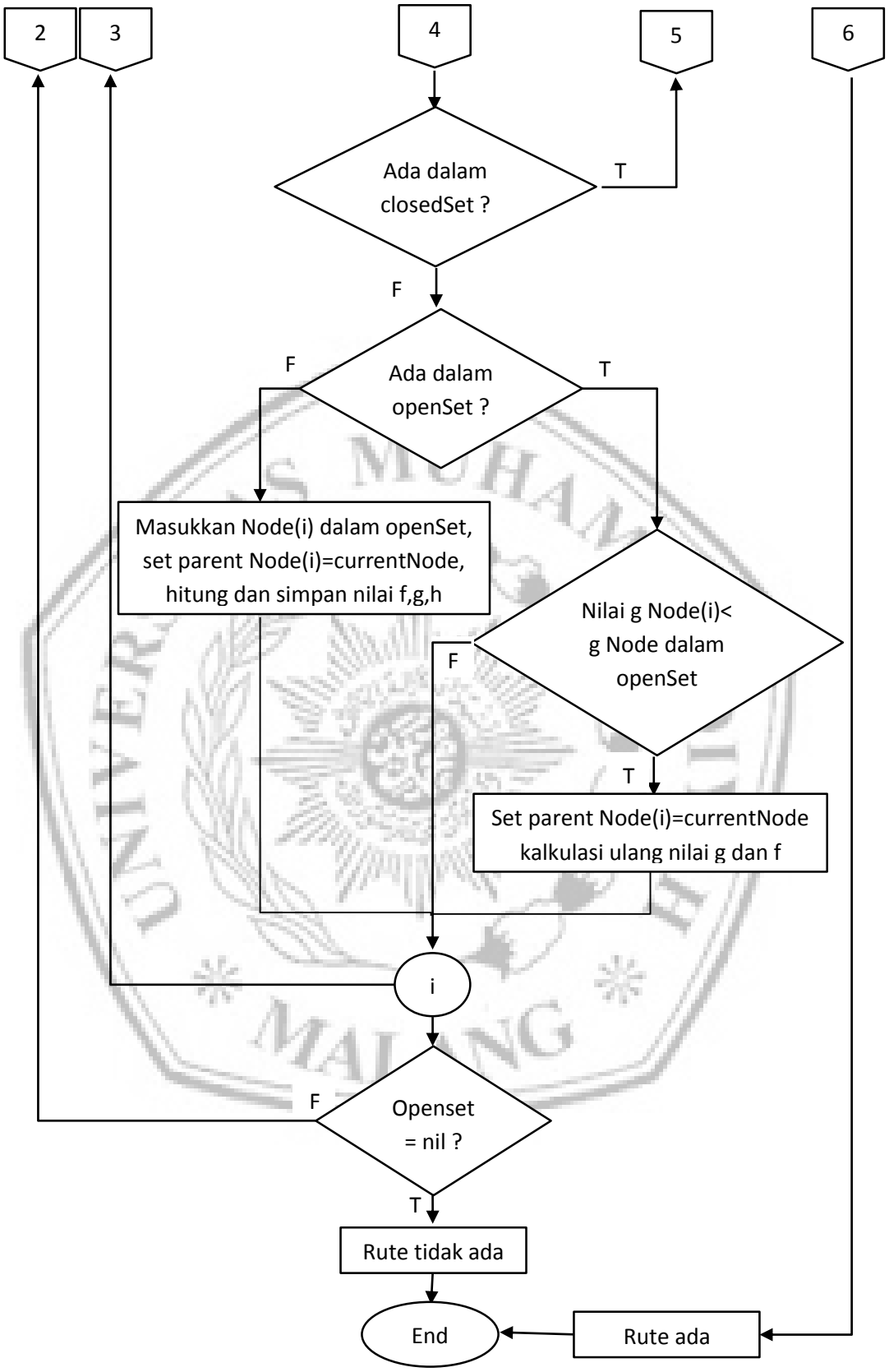


Gambar 3.1 Flowchart Game Play

b. Flowchart Algoritma A*

Berikut merupakan *flowchart* dari A* ketika proses berjalan.

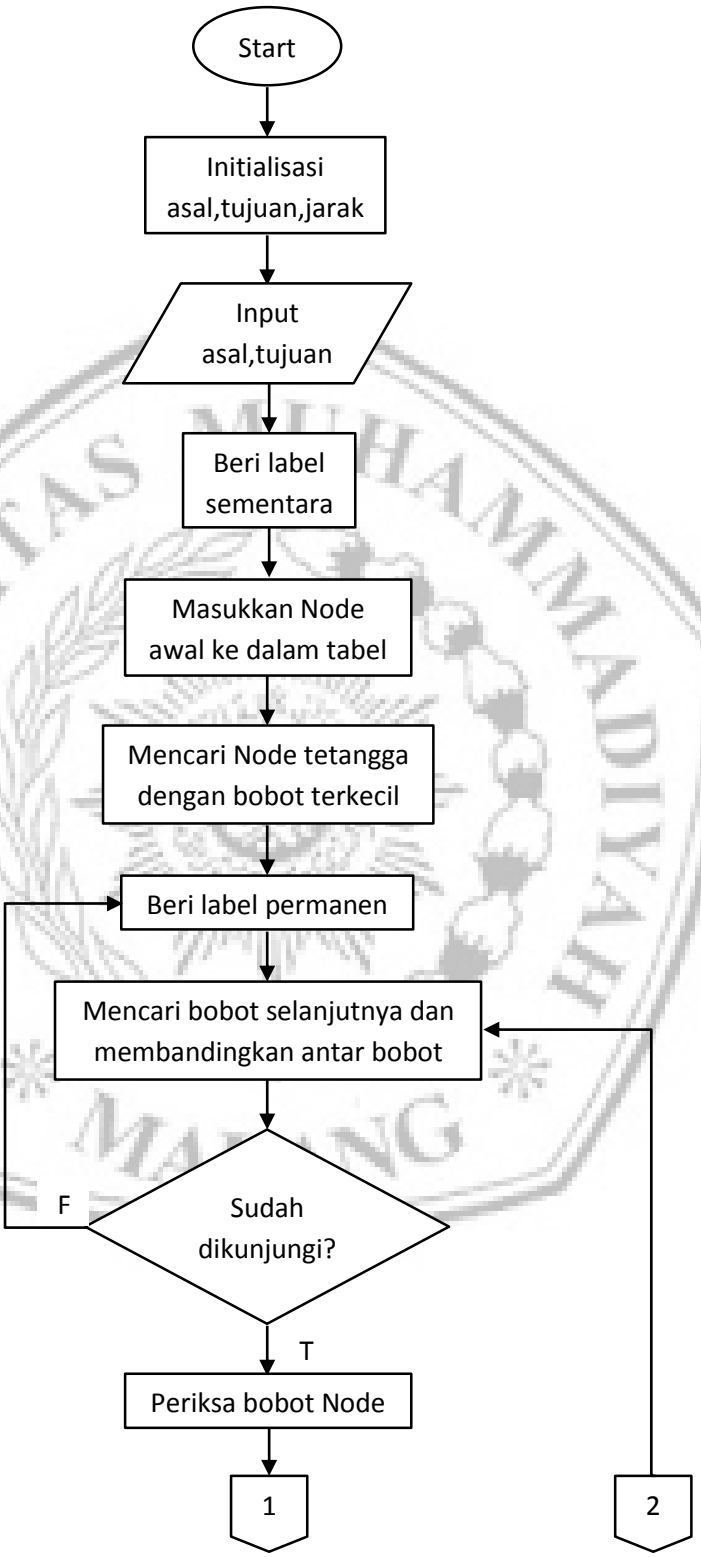


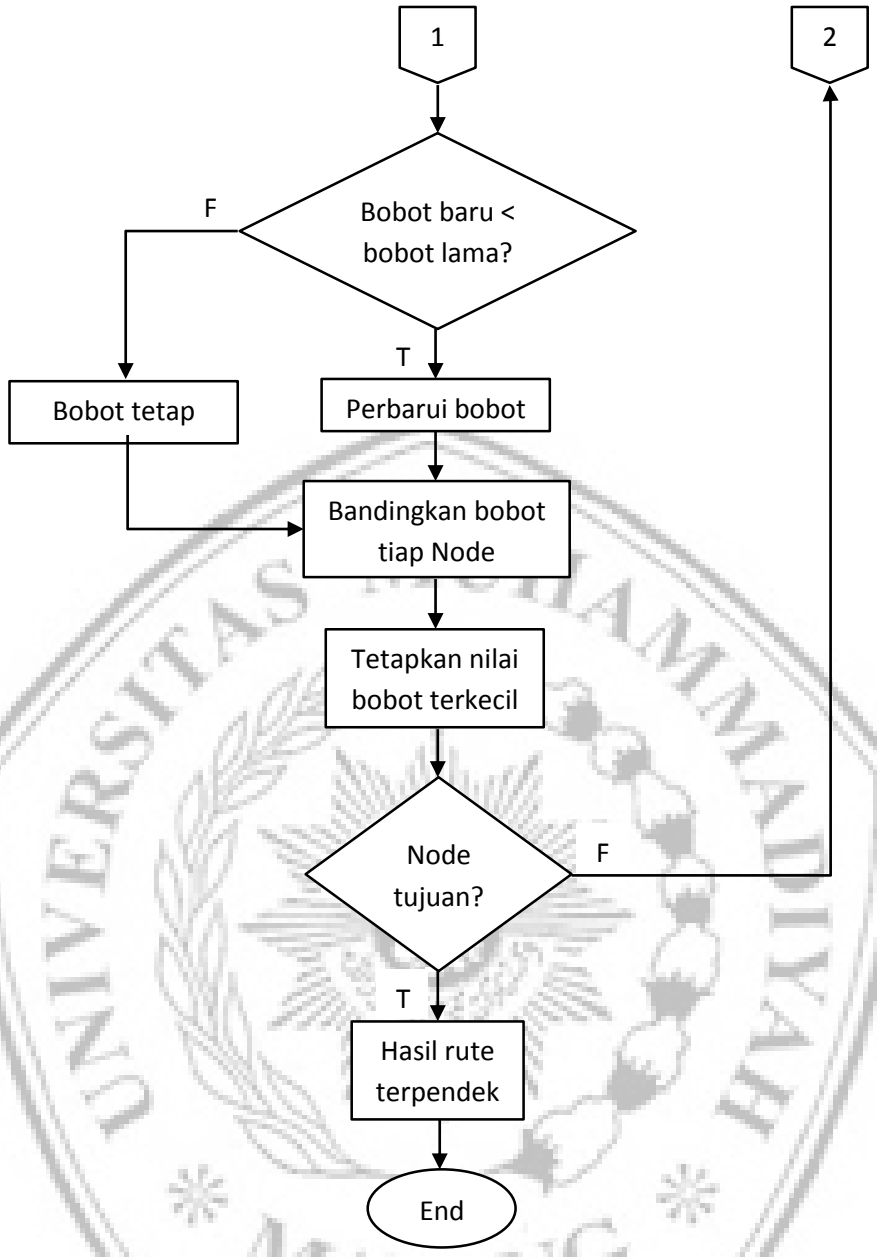


Gambar 3.2 Flowchart Algoritma *A**

c. Flowchart Algoritma Dijkstra

Berikut adalah *flowchart* dari algoritma Dijkstra saat proses berjalan.





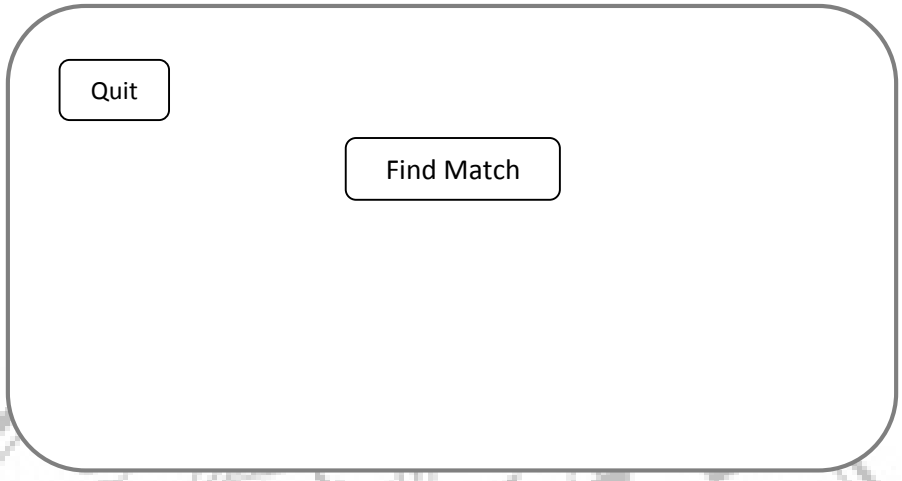
Gambar 3.3 Flowchart Algoritma *Dijkstra*

3.2.2 Desain Interface Sistem

Desain *interface* atau antar muka sistem merupakan bagian terpenting dari sistem. *Interface* dirancang untuk mengolah input dan output dari data. *Interface* juga sebagai jembatan komunikasi antara pengguna dengan sistem.

a. Form Tampilan awal

Form ini pertama kali muncul ketika *user* membuka aplikasi. Tampilan awal ini terdapat tiga ikon pilihan yaitu *Find Match*. Dengan menekan tombol *Find Match* maka pemain akan diberi pilihan untuk membuat sebuah ruang bermain atau bergabung pada ruang bermain yang telah tersedia pada pilihan *Create Game* dan *Search Game*.



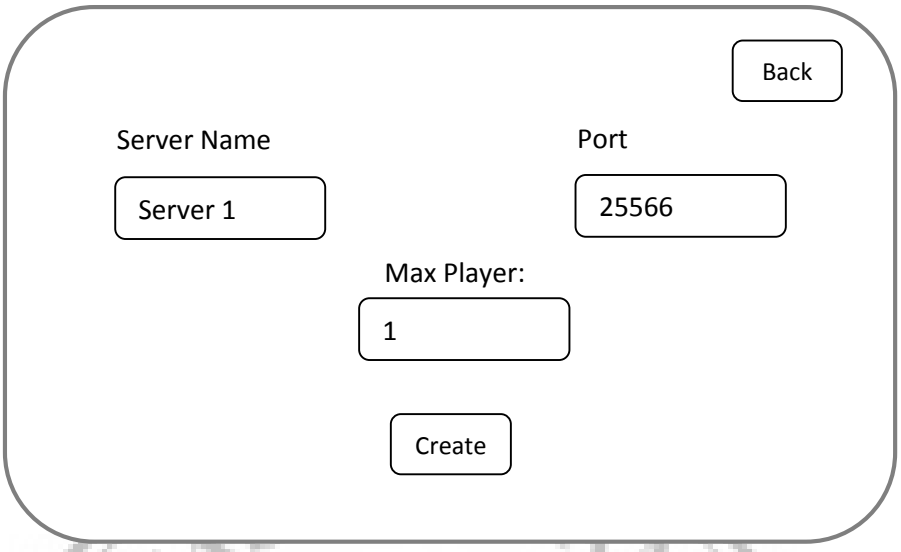
Gambar 3.4 Form Tampilan Awal

b. Form Tampilan Find Match

Gambar 3.5 Form Find Match

Pada form ini pemain dihadapkan dengan beberapa pilihan, pertama adalah *Create Game* untuk pemain membuat set game sebagai *server host.* Kedua adalah *Search Game* untuk pemain yang akan bergabung dengan *server host* sebagai *client.* Ketiga adalah *Back* untuk kembali ke menu sebelumnya. Keempat adalah *Quit* untuk keluar dari program.

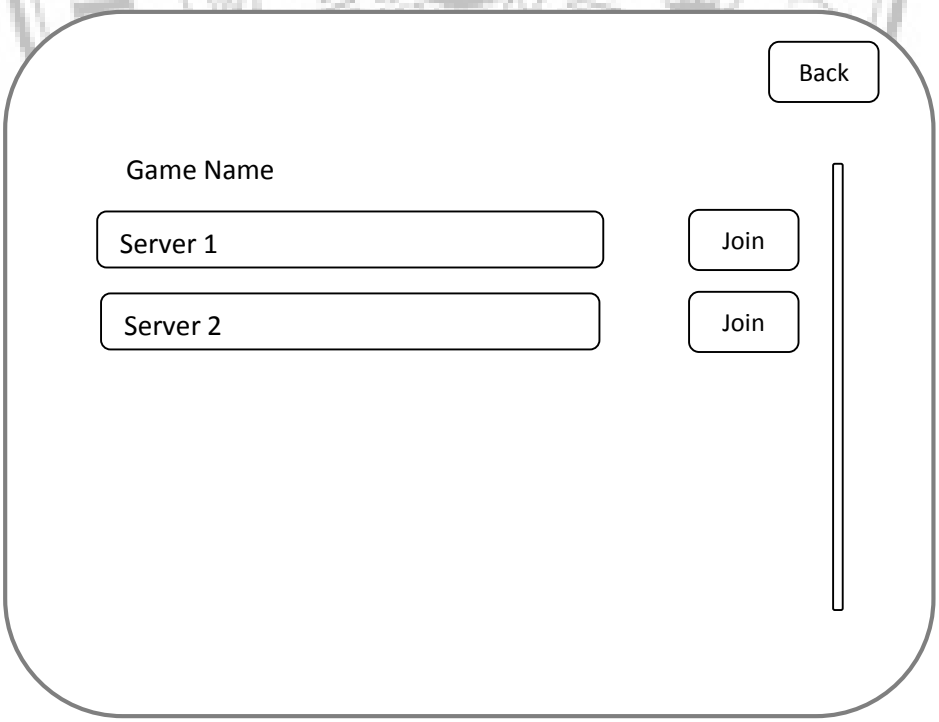
c. Form Tampilan Create Game



Gambar 3.6 Form Create Game

Ketika tombol *Create* ditekan dibuatlah satu game yang berjalan sebagai *server host* dengan Server Name dan Port sesuai yang terisi pada textfield. Untuk textfield *Server Name* digunakan sebagai nama pengguna. Pilihan lain adalah *Back*, ketika tombol back ditekan maka akan kembali ke form sebelumnya.

d. Form Tampilan Search Game



Gambar 3.7 Form Search Game

Form *Search Game* ini digunakan oleh user yang ingin bergabung pada server yang sudah ada. Dengan menekan tombol *Join* pada *Game Name* yang diinginkan maka user akan bergabung dengan game tersebut. Untuk button *Back* akan membawa player pada form sebelumnya.

UNIVERSITAS MUHAMMADIYAH MALANG